SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Lecture 4 (Moved Up): Numerical Integration and Math “Review” (Because you’ve probably forgotten it unless you had me for NWP)

Solving Ordinary Differential Equations in the Stella Environment

Agenda

- Review the Basics of the Taylor Series
- Introduce you to the numerical solving methods used in Stella

- Our Goal:
  - To make Stella less of a black box and more of a viable learning and working tool.
    - If you don't know what's going on under a GUI, you don't know what's going on – period.

What this will require in review

- Review Taylor Expansion
- The Classic Limit Theory Definition of "Derivative"

- Here we will use a very simple time-dependant relationship, y=f(y,t)

The Old Derivative's Definition

- Let's recall some basic Calc…

$$\frac{dy}{dt} = \lim_{\Delta t \to 0} \frac{y(t_0 + \Delta t) - y(t_0)}{\Delta t}$$

The Derivative's Definition

- So, if Δt is small we can approximate this so that

$$\frac{dy}{dt} \approx \frac{y(t_0 + \Delta t) - y(t_0)}{\Delta t}$$

SOUTH DAKOTA
M
SCHOOL OF MINES
& TECHNOLOGY

And playing to to solve for a y(t)?

- So using the simple definition…

$$y(t_0 + \Delta t) = y(t_0) + \frac{dy}{dt}\Delta t$$

- Which would be easy…. if dy/dt is always a constant
  - i.e., dy/dt = 6 and not 6t or $f(t^2)$ or 6cos(t)
  - But what do you do when you when you have a higher order dy/dt?

The Taylor Series

$$y(t_0+\Delta t)=y(t_0)+\frac{dy}{dt}\Delta t$$

- The Taylor Series is, for y(t) at $t=t_0+\Delta t$

$$y(t_0+\Delta t)=y(t_0)+\sum_{i=1}^{\infty}\left[\frac{(\Delta t)^i}{i!}\frac{d^i y}{dt^i}(t_0)\right]$$

$$=y(t_0)+\Delta t\frac{dy}{dt}(t_0)+\frac{(\Delta t)^2}{2!}\frac{d^2y}{dt^2}(t_0)+\frac{(\Delta t)^3}{3!}\frac{d^3y}{dt^3}(t_0)+\dots$$

- So… the more you know about the higher order terms of the relationship y(t) as it changes, the better you can project its value onto future values of *t* from a known starting point, y(t_0) at time t_0

The Taylor Series

- FOR YOUR NOTEBOOK!

$$y(t_0+\Delta t) = y(t_0) + \sum_{i=1}^{\infty}\left[\frac{(\Delta t)^i}{i!}\frac{d^i y}{dt^i}(t_0)\right]$$

$$= y(t_0) + \Delta t\frac{dy}{dt}(t_0) + \frac{(\Delta t)^2}{2!}\frac{d^2 y}{dt^2}(t_0) + \frac{(\Delta t)^3}{3!}\frac{d^3 y}{dt^3}(t_0) + \ldots$$

- Get the following… for the 0th through 5th order.

$$y(t) = t^3$$

$$t_o = 1;\quad y(t_o) = 1;\quad \Delta t = 2;\quad y(t=3) = ?$$

- Answer: 1, 7, 19, 27, 27, 27, 27, 27, …..

The Taylor Series in Modeling

$$y(t) = t^3$$

$$\frac{dy}{dt} = 3t^2$$

- Applying it to modeling…
- We know that at t=1, $y(t=1)=1^3=1$
- If we knew the perfect solution to y(t), which here is t^3, then we'd know that $y(t=3)=3^3=27$

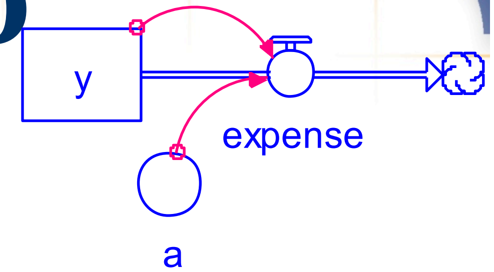
- But… we don't know the perfect solution… otherwise modeling would be a perfect and *precise* art.

The Taylor Series in Modeling

- We DO, however, hopefully know what forces are acting on our system at our current point in time (or how to guess at them)
- Therefore we can approximate the potential changes to our system NOW, and project them into the future, based on what we know NOW – assuming that changes in the system are modest over delta-t.
- And the better we can approximate these potential changes, the better the solution!

Spiraling Down to Concretes

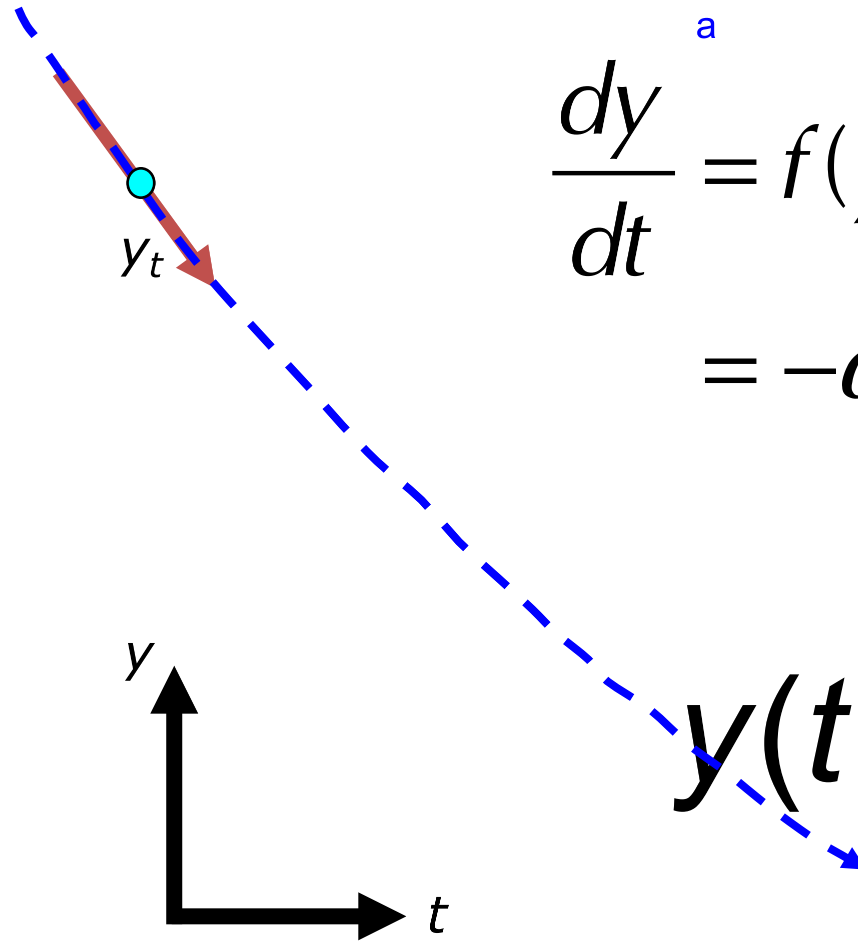
$$\frac{dy}{dt}(t) = -\alpha y(t)$$



- Consider the following blue curve. The "perfect" real solution.
- This is the system we want to model.
- We are at the light blue point.
- We want to project the future position of the point along the "perfect" curve.

$$\frac{dy}{dt} = f(y,t)$$

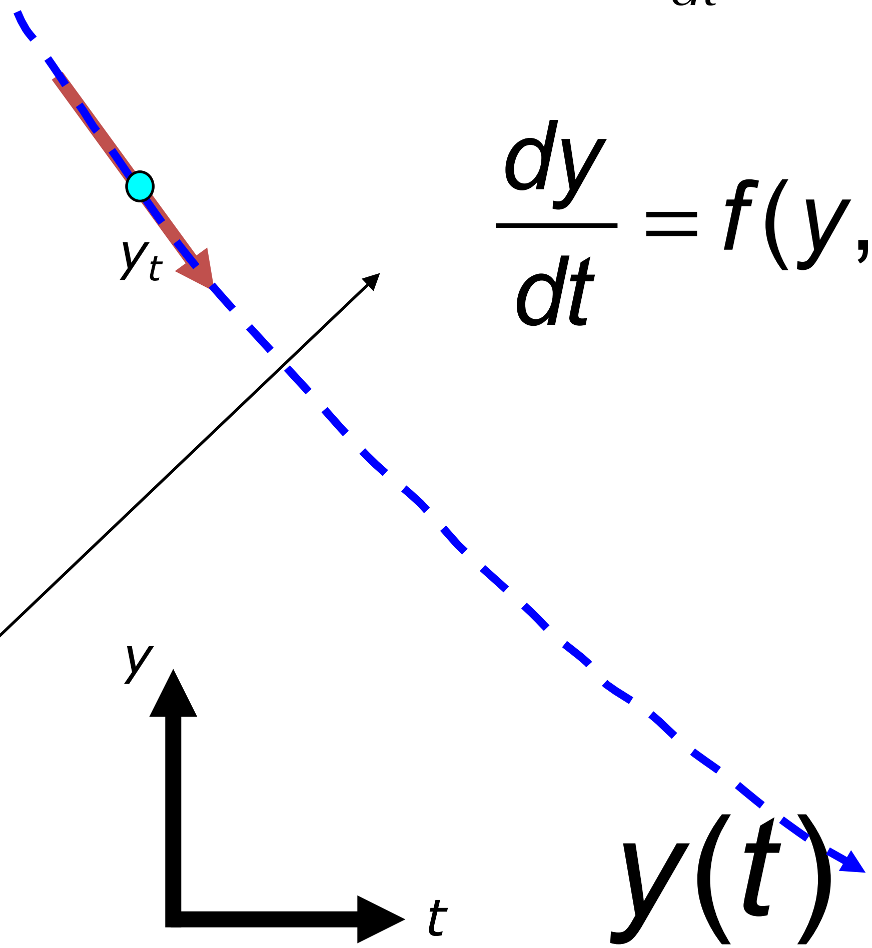
$$= -\alpha y$$



Spiraling Down to Concretes

$$\frac{dy}{dt} = c_1 - c_2 y^4$$
$$\frac{dy}{dt} = -\alpha y$$

- Before proceeding let's sit back and look at a given system.
- We have a parameter we wish to solve (or model): y(t).
- We begin by <u>*not*</u> attacking the formula for y and "solve" y



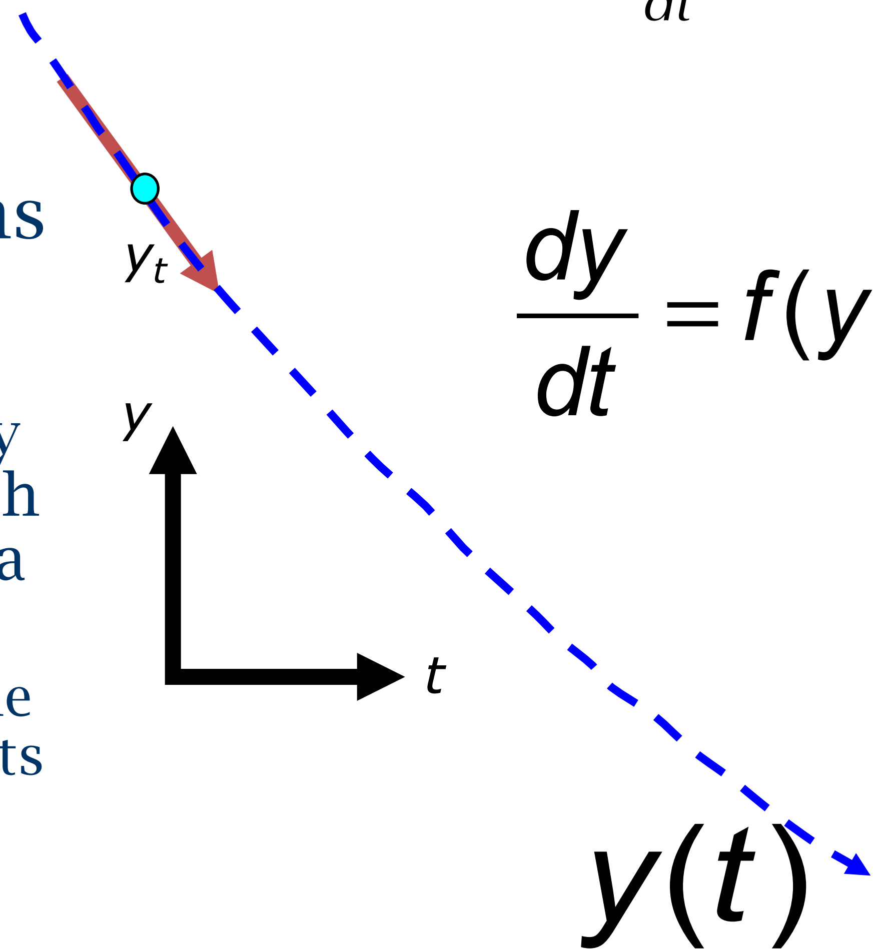
$$\frac{dy}{dt} = f(y,t)$$

Spiraling Down to Concretes

$$\frac{dy}{dt} = c_1 - c_2 y^4$$

$$\frac{dy}{dt} = -\alpha y$$

- We begin by <u>*not*</u> expressing the formula for y
- Rather, we express y in terms of its ***changes***.
  - In our example, y could be temperature or internal energy of the earth (or the water depth of a bucket – or the dollars in a trustfund…)
    - Its change is a function of the incoming radiation (c_1) and its ability to offload energy ($c_2 T^4$)

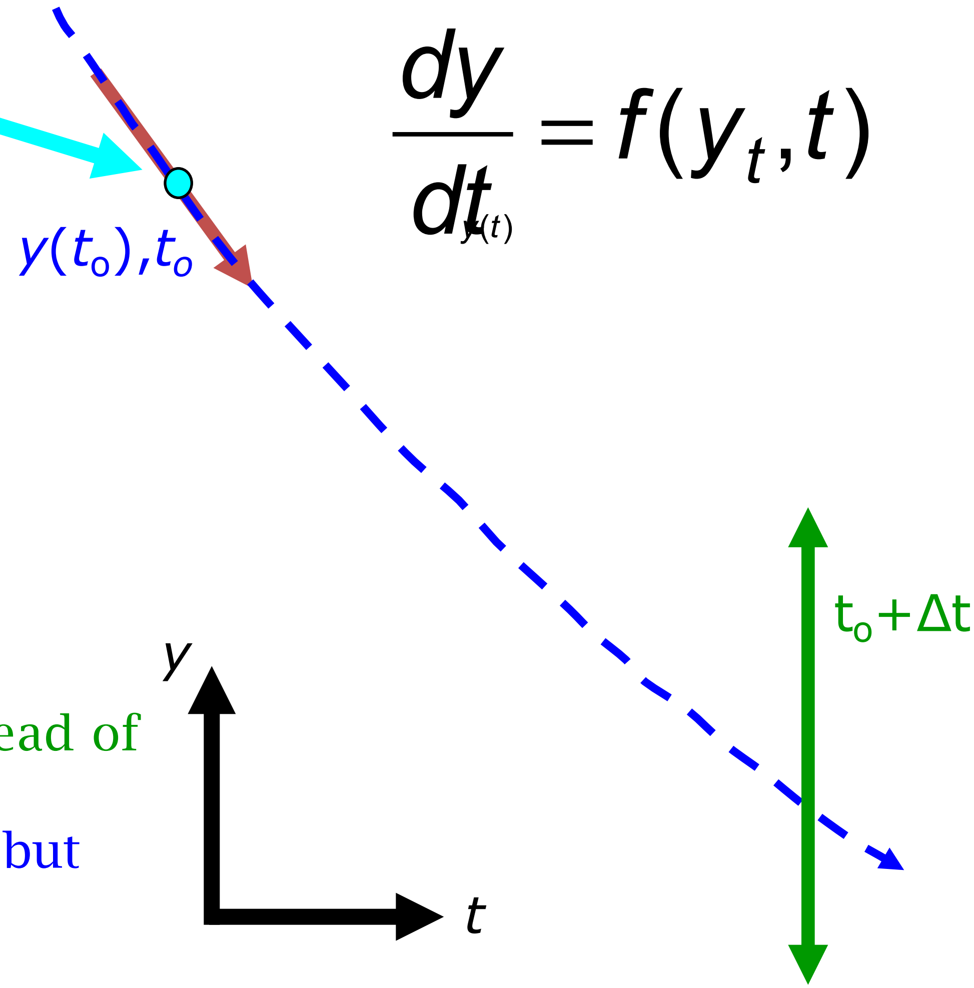


$$\frac{dy}{dt} = f(y,t)$$

Spiraling Down to Concretes

- “You are here”
  - At the Carolina Blue Dot
  - $t = t_0$
  - $y = y_0$
  - $dy/dt = f\,[y(t_o), t_o]$
  - These values we know
- “We want to be there”
  - Green Line
  - to+Δt
  - y=?
    - hence the vague line instead of a green dot
    - (the blue line is the ideal but still an unknown!)

$$\frac{dy}{dt} = f(y_t, t)$$



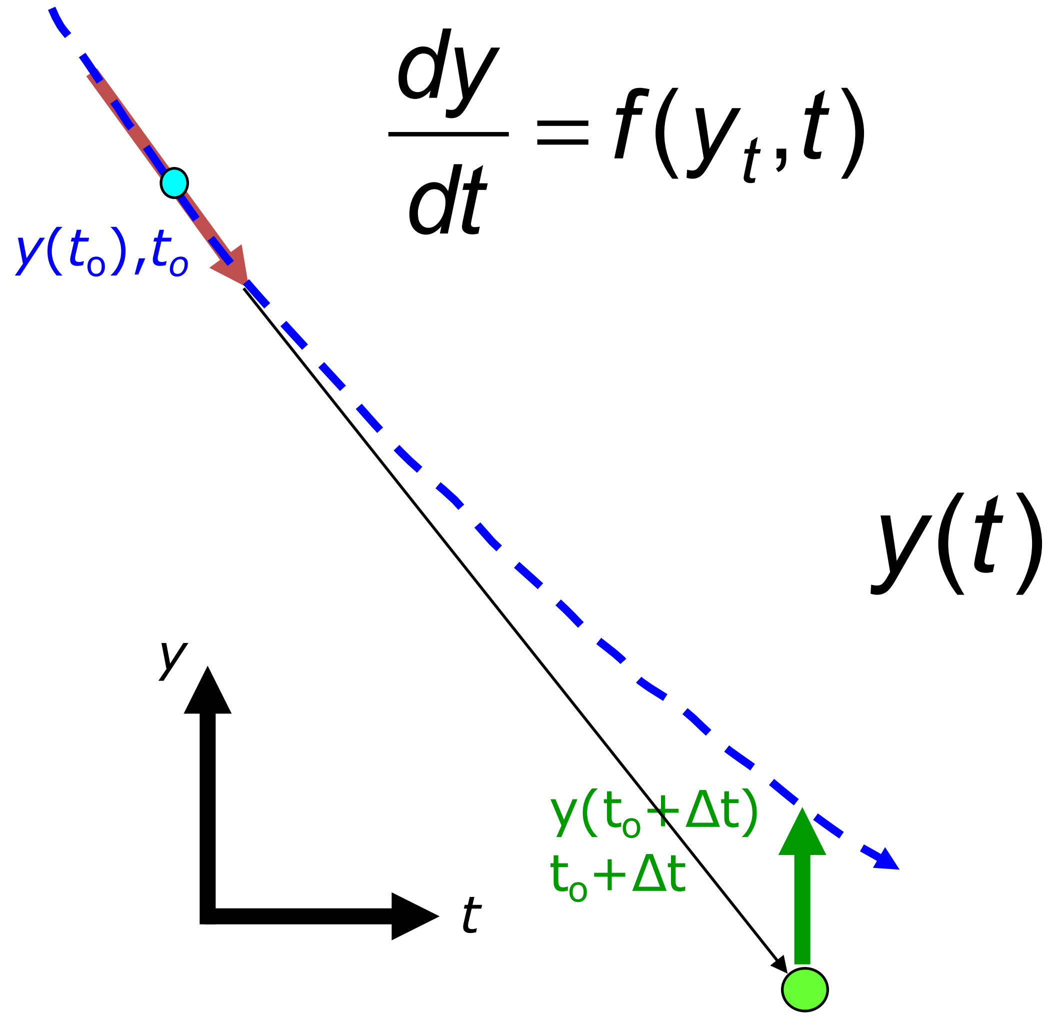
Spiraling Back Down to Concretes

- Notice what happens if you take the first derivative approximation and solve for $x(t_0+\Delta t)$

$$y(t_0+\Delta t) \approx y(t_0) + \Delta t \frac{dy}{dt}(t_0)$$

- …same as the 1rst-Order approx of the Taylor Series
  - (This is also a variant on the same eq. that I tossed about periodically last time and is what stella "writes" for you)

$$\frac{dy}{dt} = f(y_t, t)$$

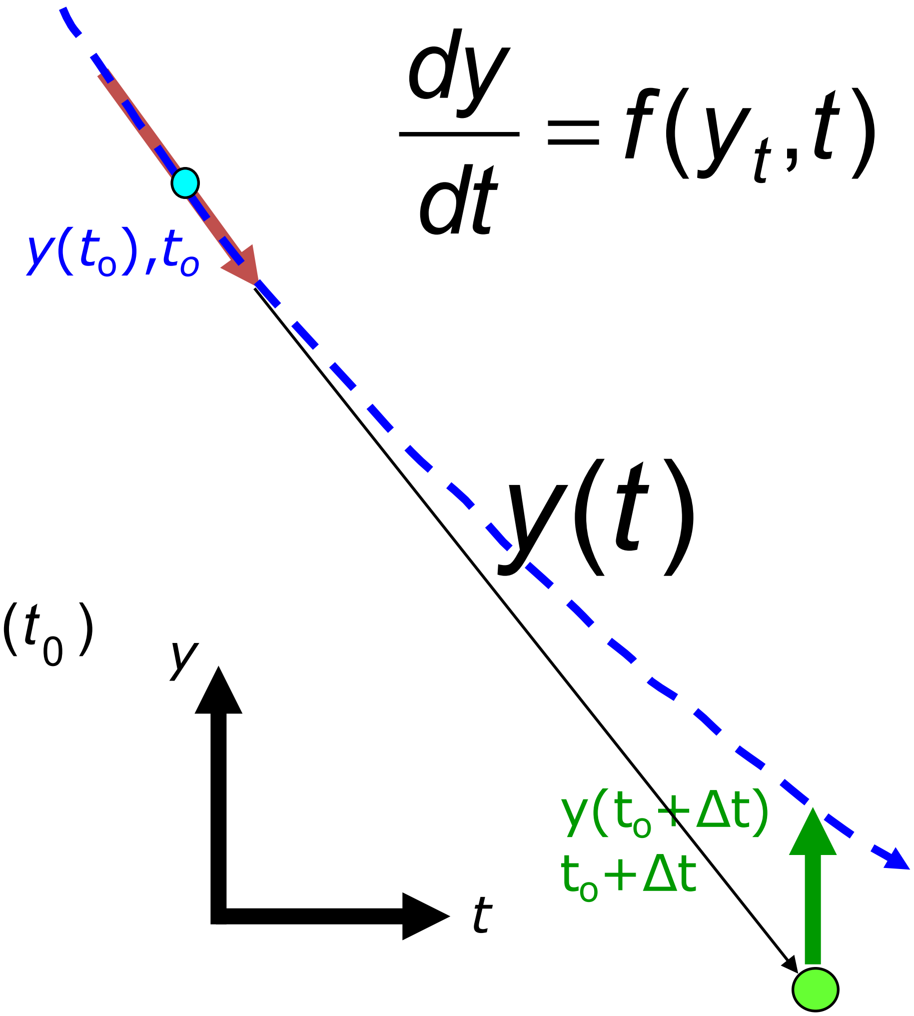


Spiraling Back Down to Concretes

- This solution approach is called ***Euler's Method***. (not to be confused with Euler's theorem or Eulerian expansion) and can serve as a fast solution approach.

$$\frac{dy}{dt} = f(y_t, t)$$

$$y(t_0 + \Delta t) \approx y(t_0) + \Delta t \frac{dy}{dt}(t_0)$$

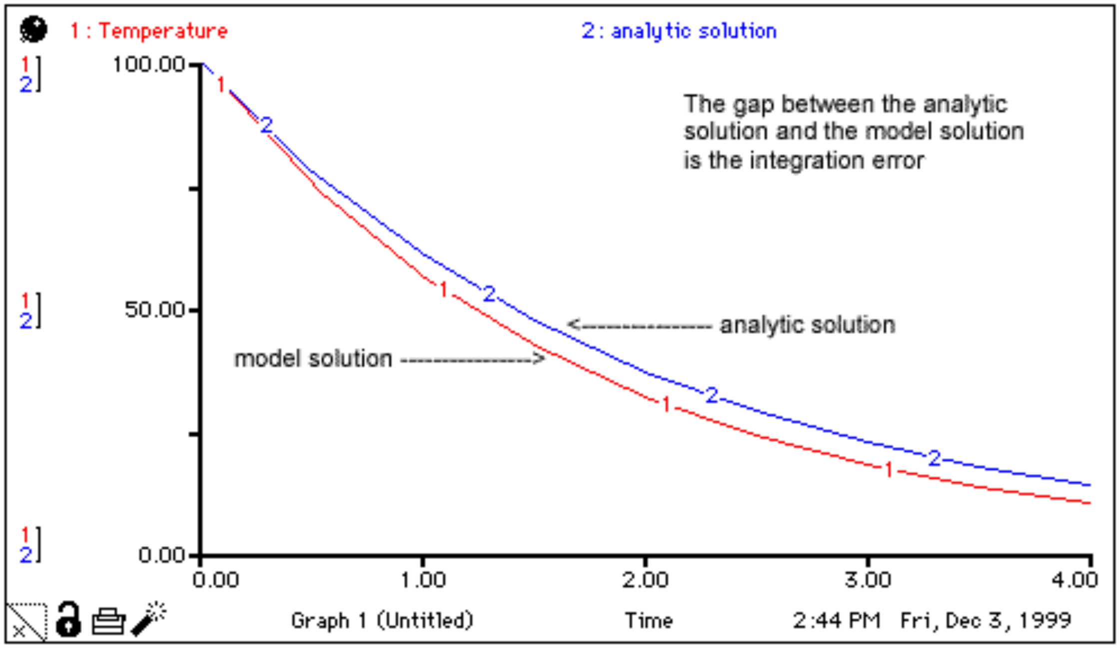


Euler's Method

- Euler's method is clearly very simple and straight forward.
- Unfortunately, it is not the most accurate of methods.
- *It is reliant on the* ***first derivative*** *at the* ***point of origin*** *for a given time step*.
  - Analogy: "Driving faster than the range of your headlights"
  - If delta-t is big, you're gonna accrue error.
- Consequently, the smaller the time step the better the prediction, but error is not fully eliminated.

Euler’s Method

- To the side is an other example:
  - Classic Cooling
- The Blue indicates an analytical solution to the equation.
- The Red, a Euler-estimated approach.
- (This works better in a spreadsheet than the previous example)



$$\frac{dT}{dt} = -A_1 T$$

$$T = A_2 e^{-A_1 t}$$

Runge-Kutta Method

- Let's then look at another approach:
- The ***Runge Kutta Method***. This method provides superior accuracy than Euler's Method and can handle larger time steps as well.
- But before we proceed, let's change some notation.

- This is mostly to reduce clutter and to make what we do here compatible with most textbooks and refernce manuals like *Numerical Recipes*

Notation Change

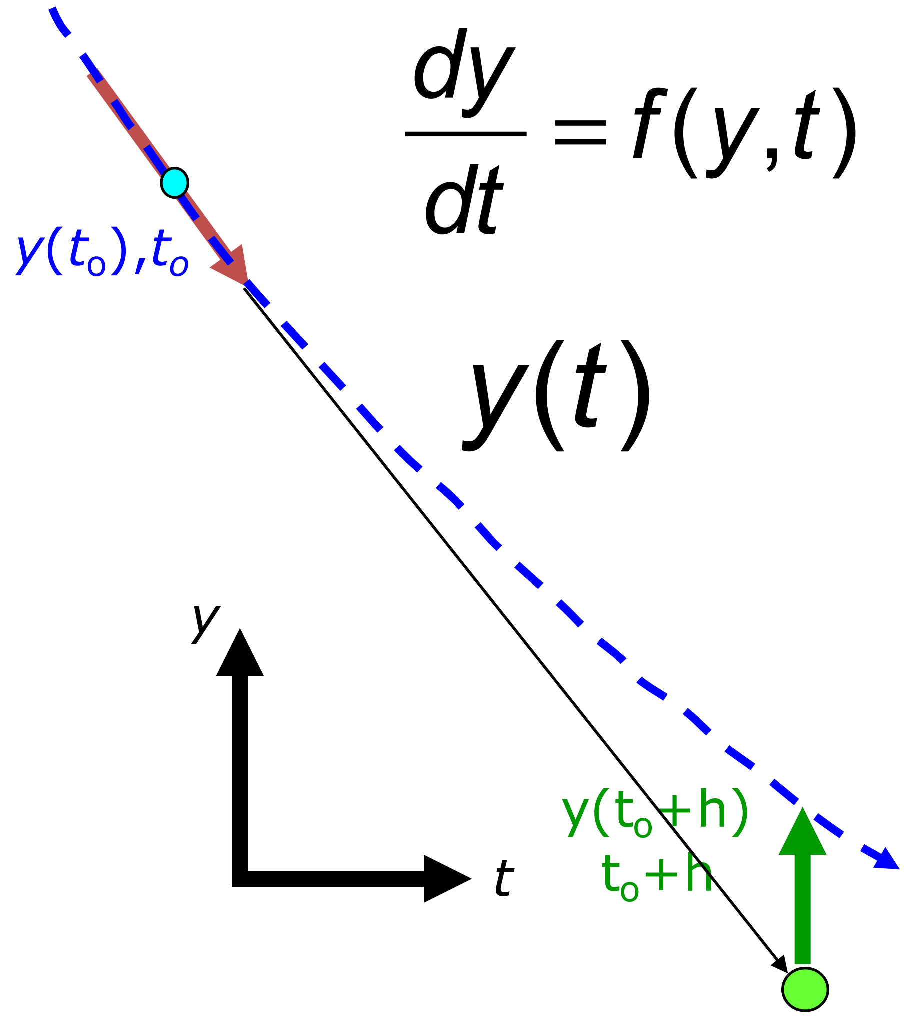
- We're going to use "f" for the first derivative
  - *"f" for "forcings"*

$$\frac{dy}{dt} = f(y,t)$$

$$y(t_0 + \Delta t) \approx y(t_0) + \Delta t\, f[y(t_0), t_0]$$

also

$$h \equiv \Delta t$$



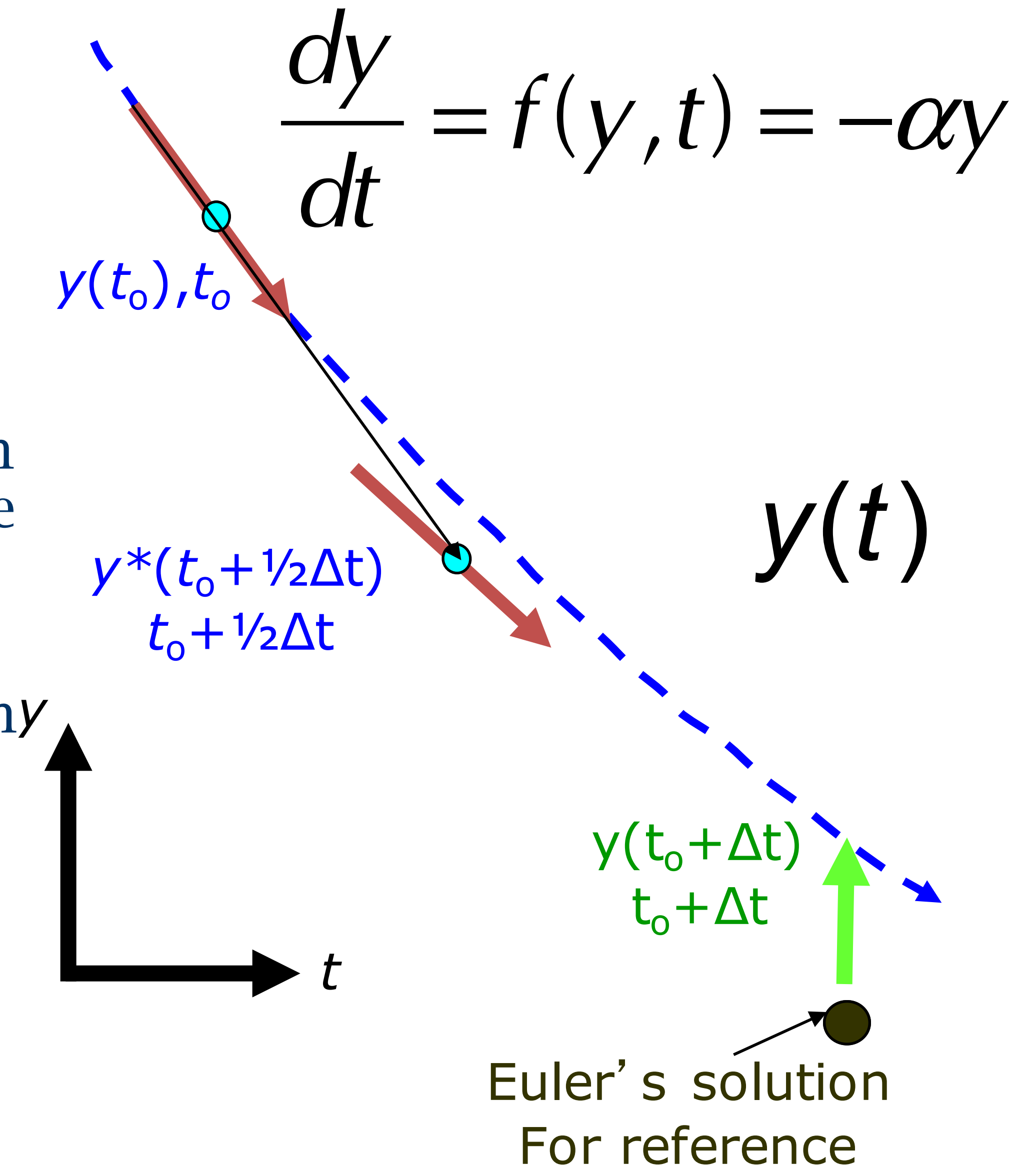
Runge-Kutta Method

- If we were to take an incremental approach, we could define our derivatives at intermediate values between our steps…

- Today, we'll look at the 1rst-, 2nd- and 4rd-Order R-K approaches

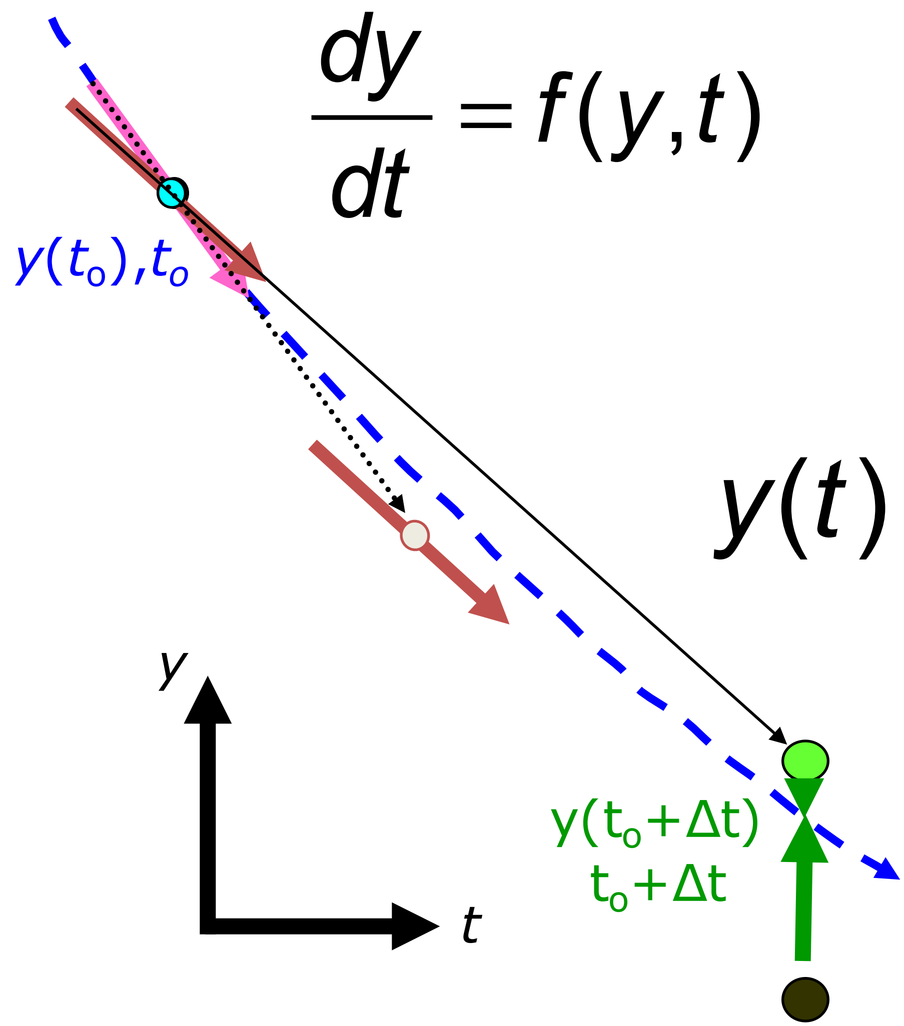
2^{nd}-Order Runge Kutta

- Let's shoot *only* half-way first to the mid-point between t and t+Δt
- We now have an intermediate slope between the two points.
- This "midpoint" progression allows us to aim now for the next step.
- ***But*** in case we are "off" with our y estimate, we'll hedge our bets and shoot again from t_o.



2^{nd}-Order Runge Kutta

- Using this new slope…
  - Notice, first, that we assume a continuous curve here
- … We re-cast our line back to t=t+Δt for our final estimate



2^{nd}-Order Runge Kutta

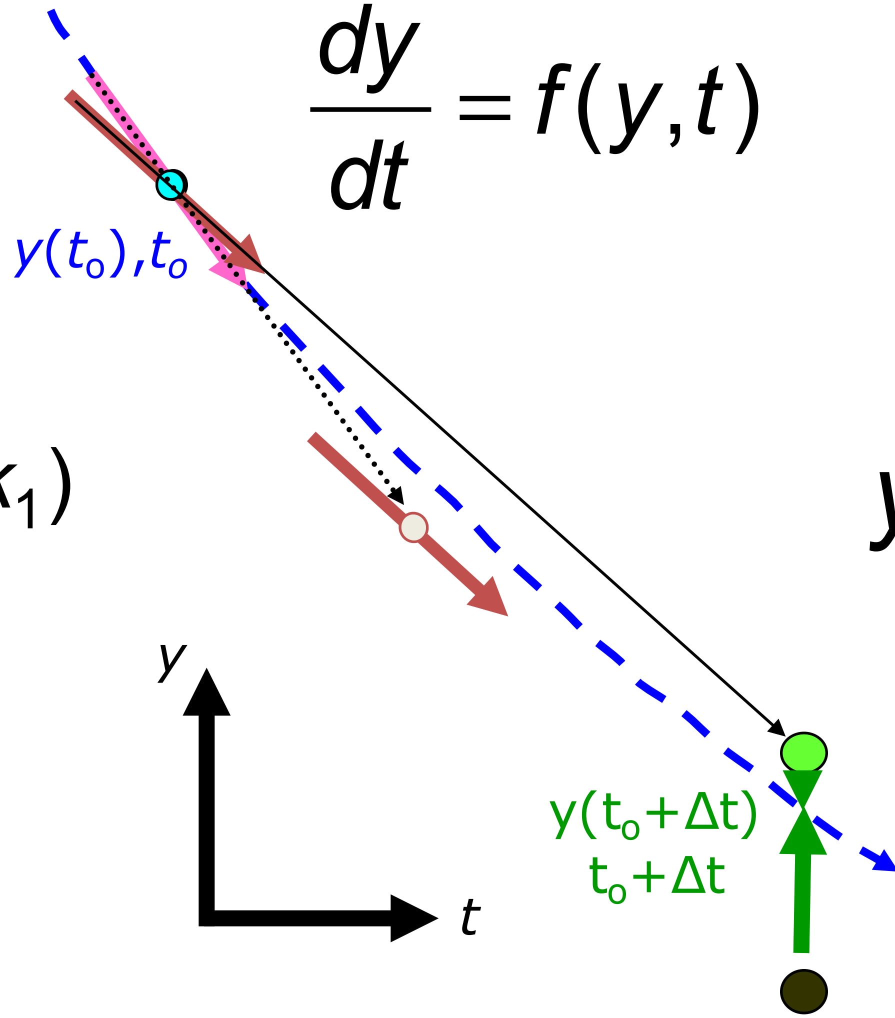
- In the world of perfect forms (aka textbooks):
  - Since Fortran doesn't do pictures!

$$\frac{dy}{dt} = f(y,t)$$

$$y_{n+1} = y_n + k_2 + O(h^3)$$

$$k_2 = h\, f(t_n + ½h, y_n + ½k_1)$$

$$k_1 = h\, f(t_n, y_n)$$

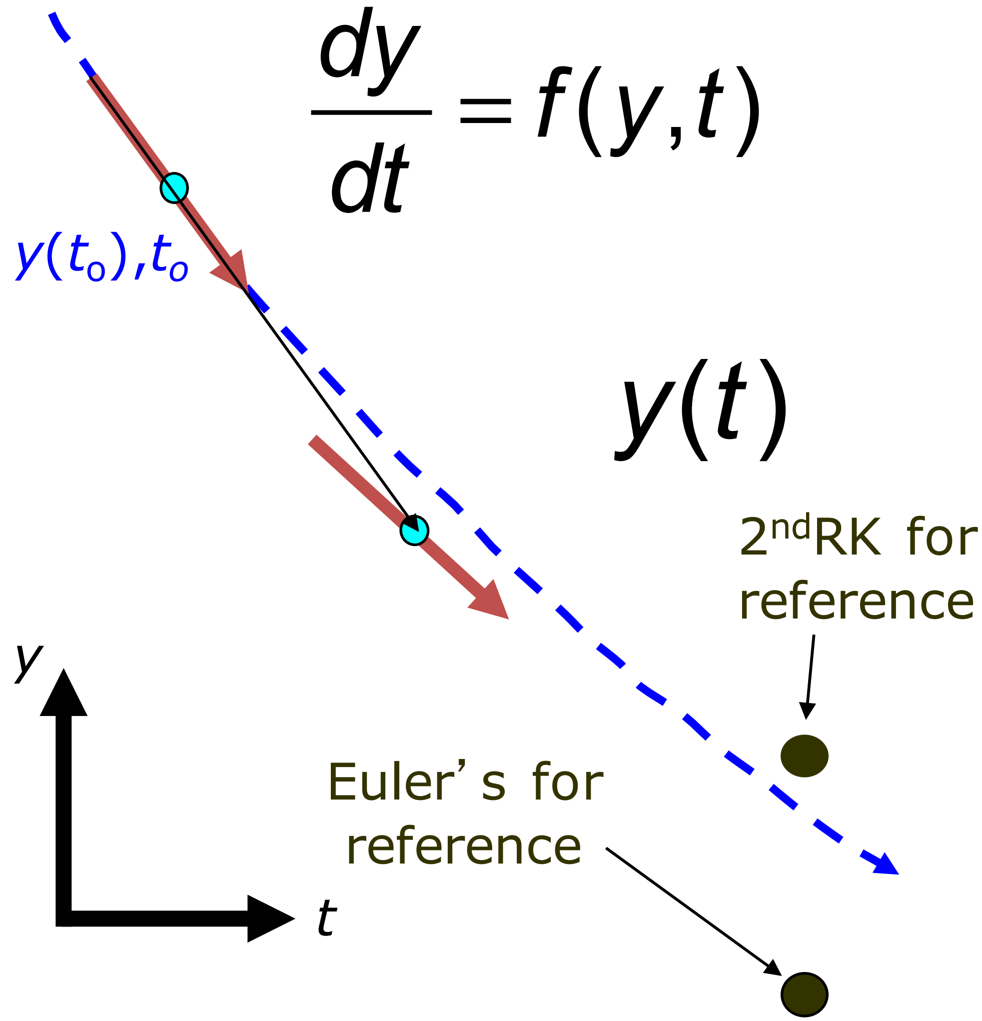


- This approach is also called the "mid-point" method

4^{nd}-Order Runge Kutta

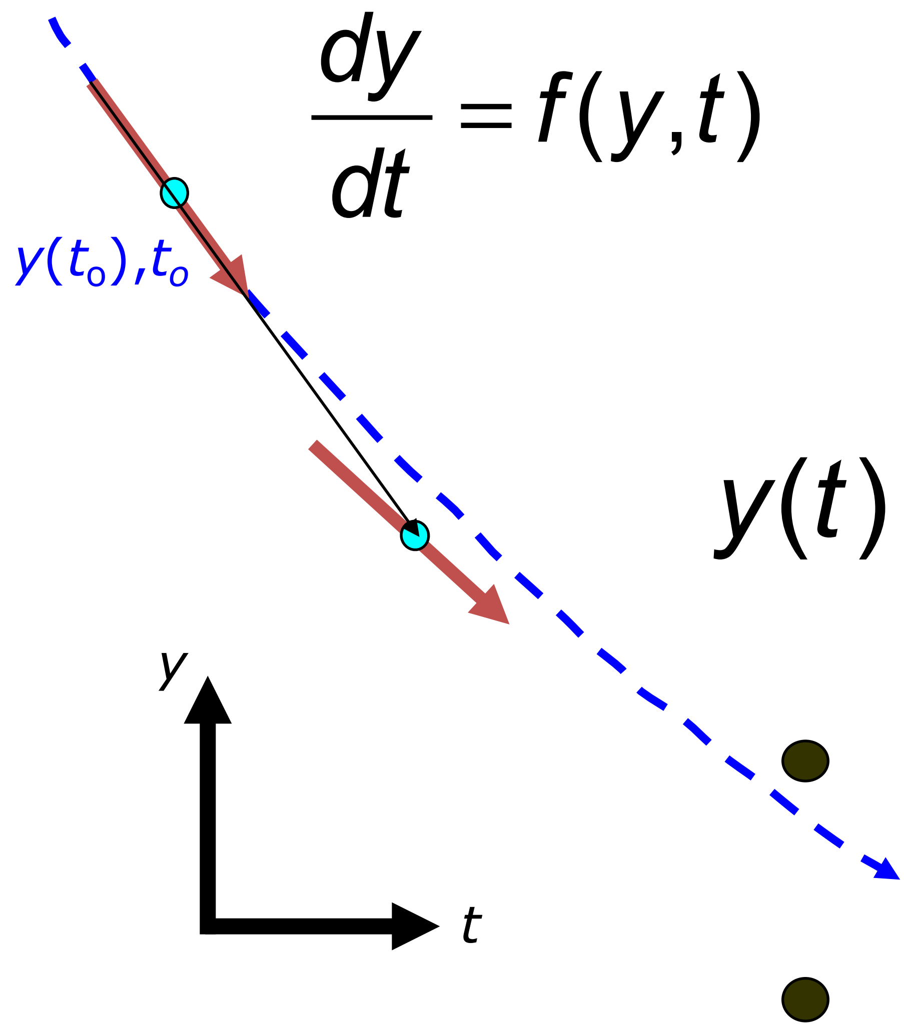
- We can expand this to a 4^{th}-order RK (the first order, btw is the Euler's method (if you go backward from the 2^{nd}-order example, you might be able to see why).
- But for the 4^{th}-order approch we will cast our line more than once (actually, 4 times)

$$\frac{dy}{dt} = f(y,t)$$



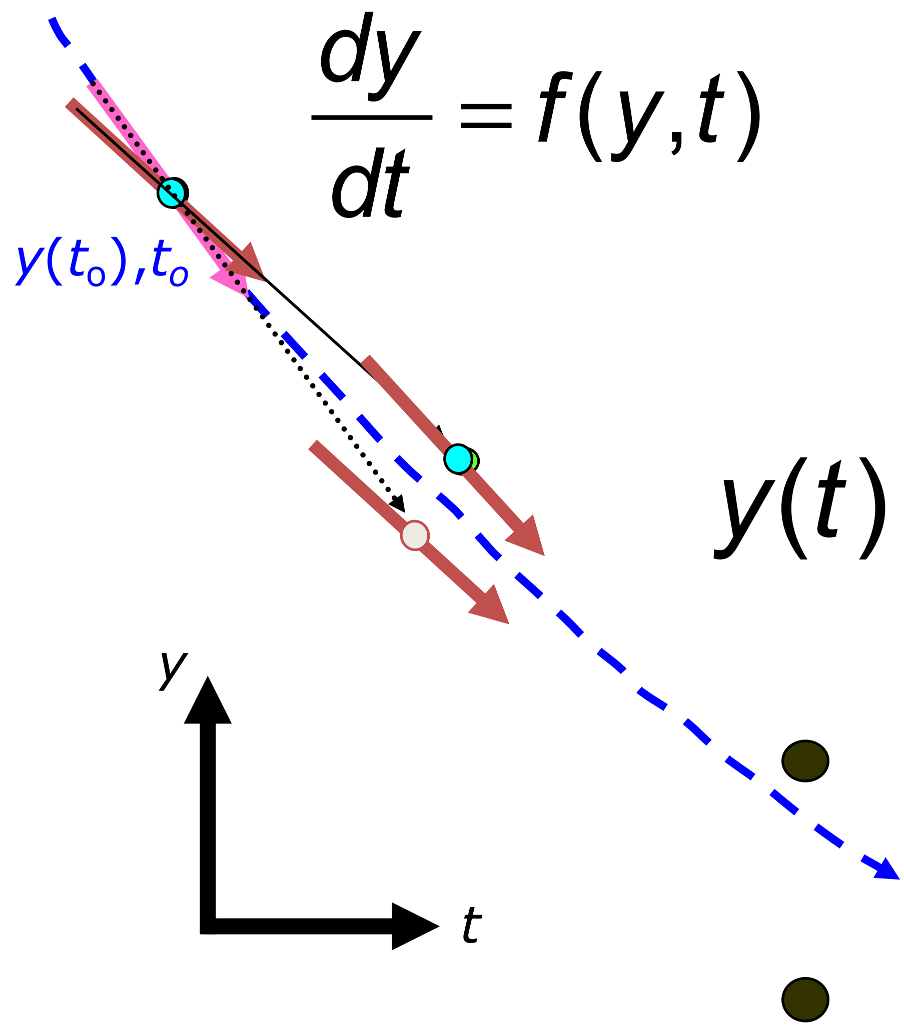
4^{nd}-Order Runge Kutta

- Cast Number 1
  - The first cast is the same as the 2^{nd}-order.
  - From here, we will cast our second line, not all the way to $t+\Delta t$
  - As with the first cast, we'll send it only half way!



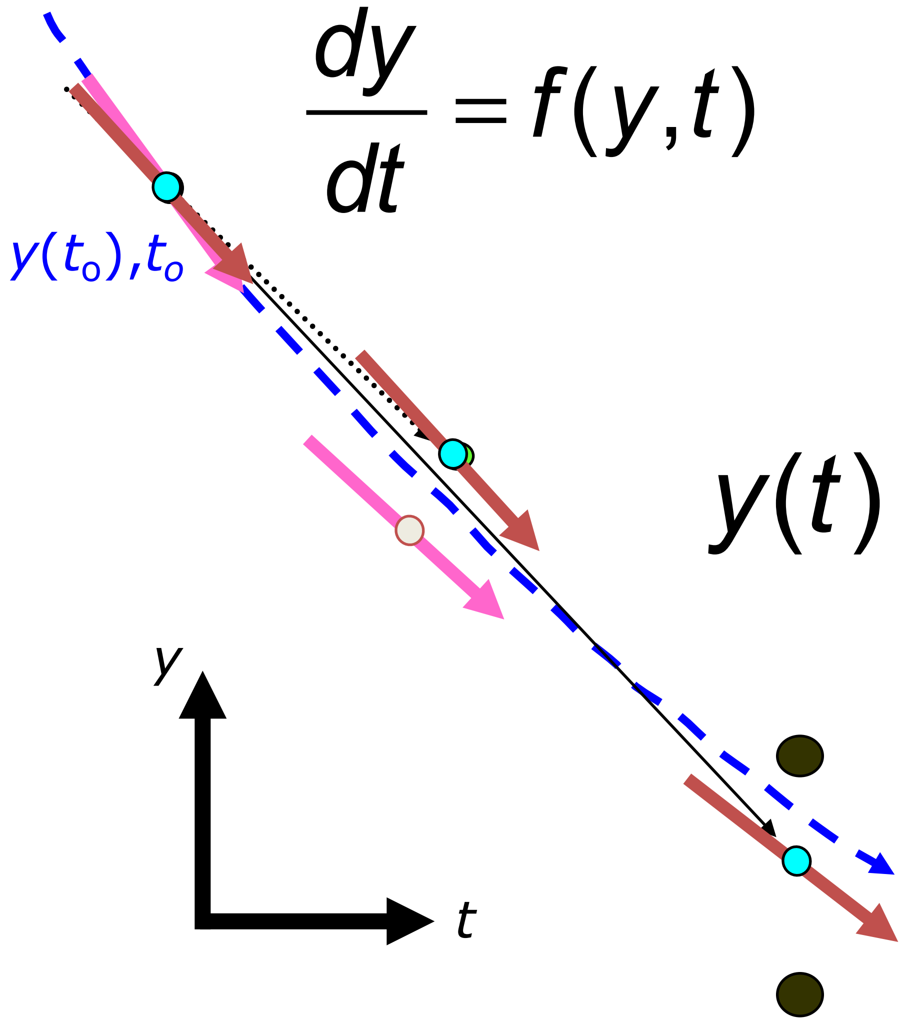
4nd-Order Runge Kutta

- Cast Number 2
  - Notice that even though this new slope is measured at the same time as the last one, its value can differ from the first "cast"



4nd-Order Runge Kutta

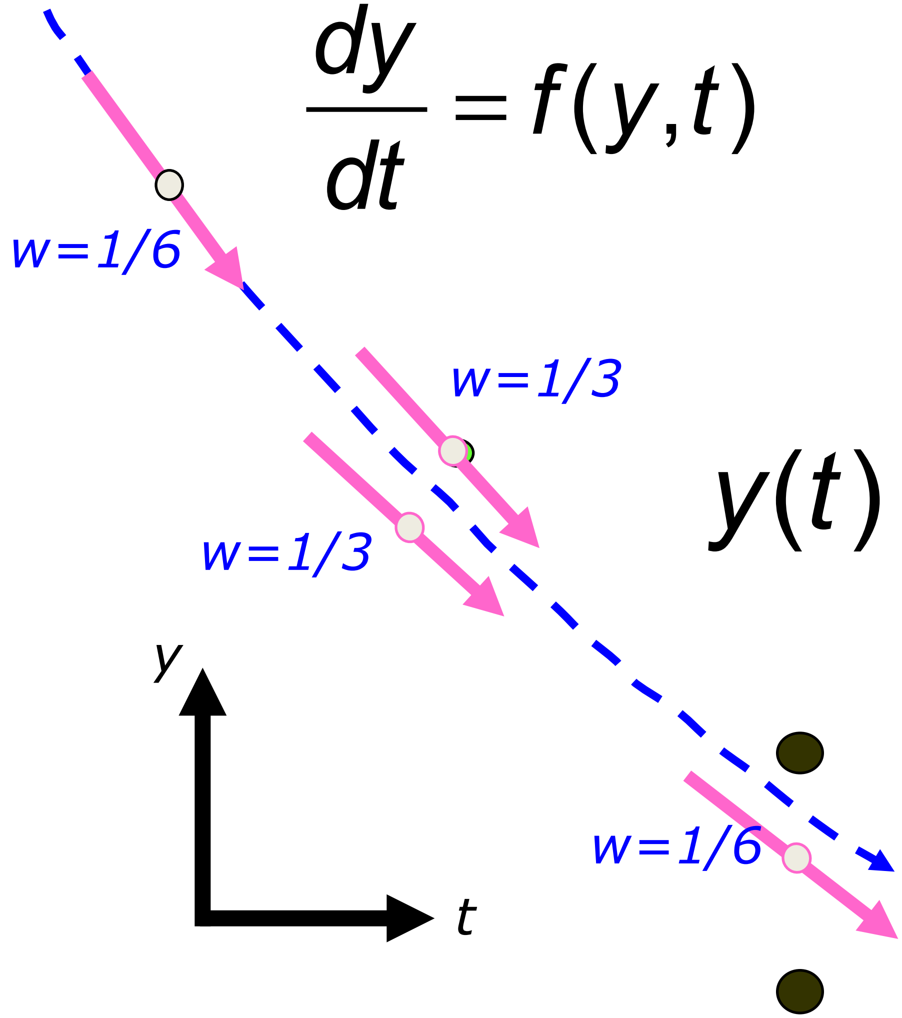
- Cast Number 3
  - Ok, this time we go all the way – but we won't be done yet.
  - Now we have FOUR possible forcings on our system over the full timestep



4^{nd} Order Runge Kutta

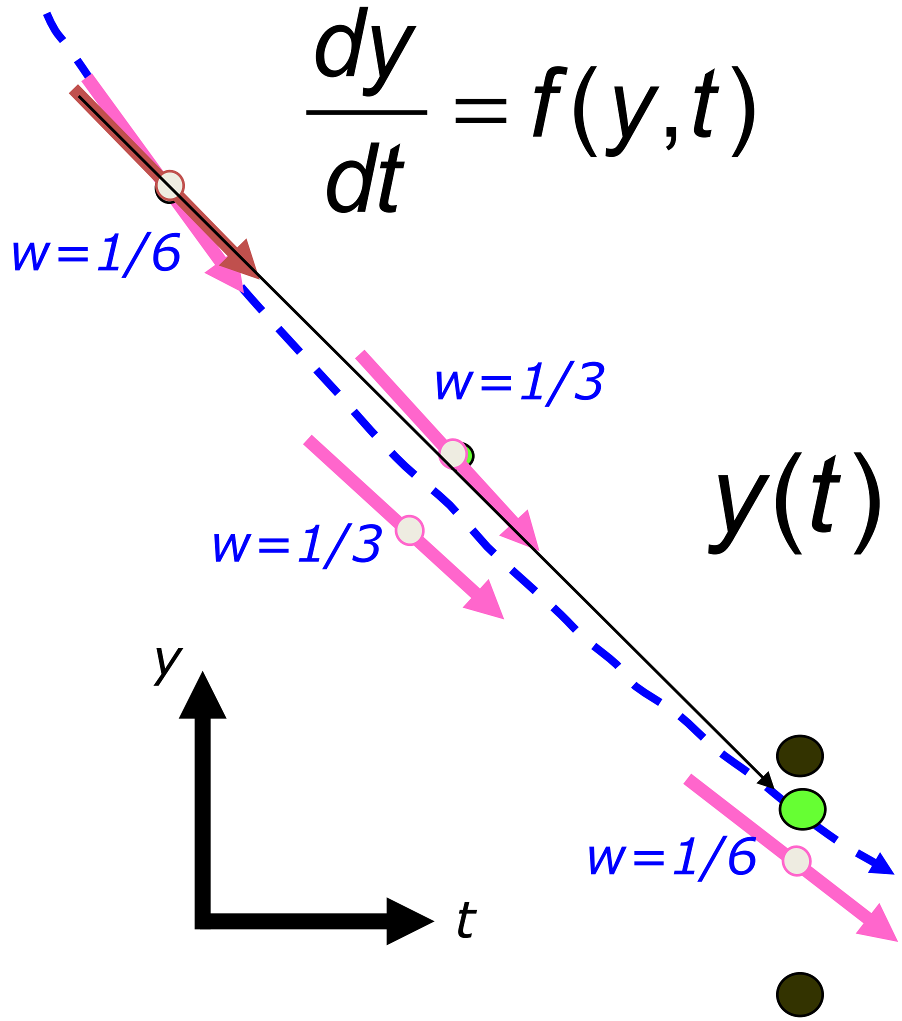
- Creating Cast 4
  - We now have four slopes (possible "forcing trajectories")
  - We'll now take a weighted average of the four noting the larger potential for error for the first- and last-cast slopes

$$\frac{dy}{dt} = f(y,t)$$



4nd-Order Runge Kutta

- Creating Cast 4
  - We how have one last cast, this time "aiming" with the weighted mean of the possible forcings.



4^{nd}-Order Runge Kutta

- Back to the math

$$\frac{dy}{dt} = f(y,t)$$

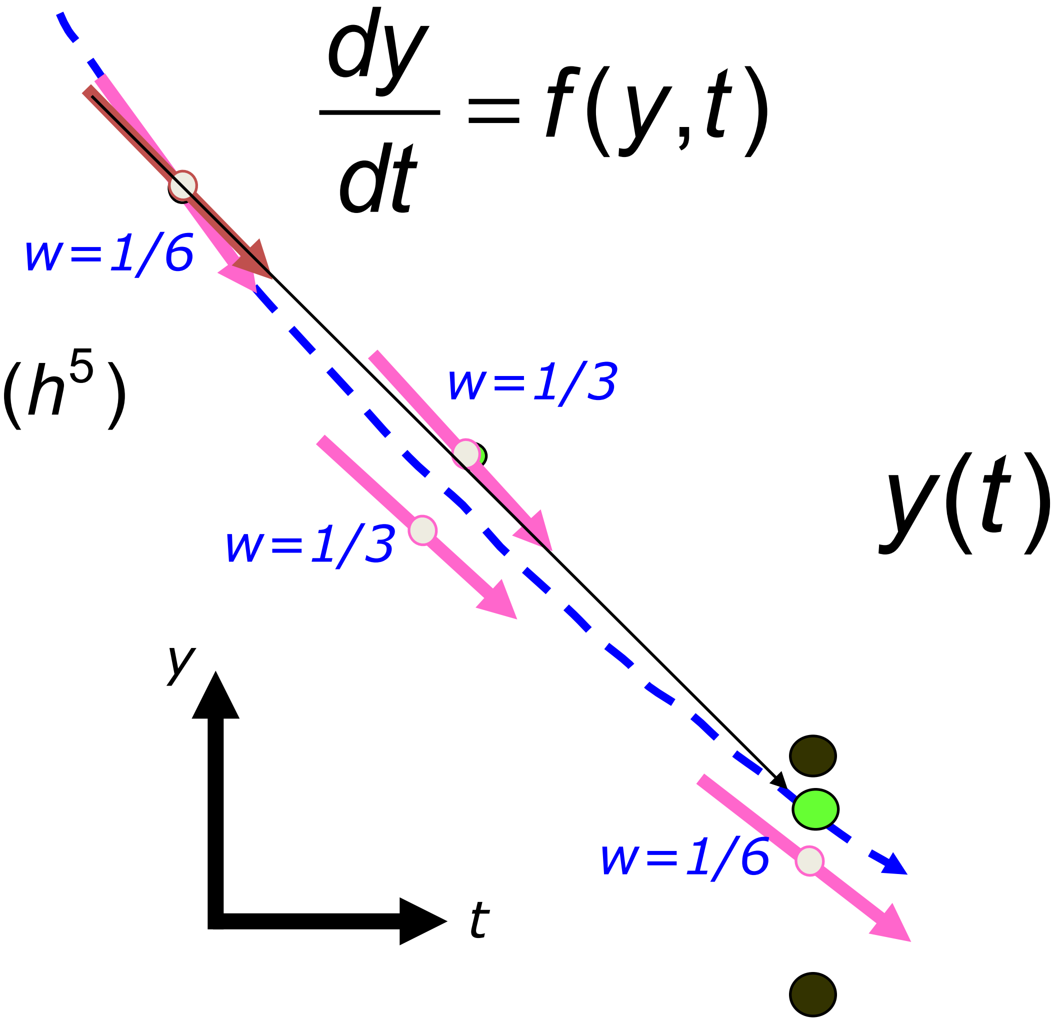
$$y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(h^5)$$

$$k_4 = h\,f(t_n + h, y_n + k_3)$$

$$k_3 = h\,f(t_n + ½h, y_n + ½k_2)$$

$$k_2 = h\,f(t_n + ½h, y_n + ½k_1)$$

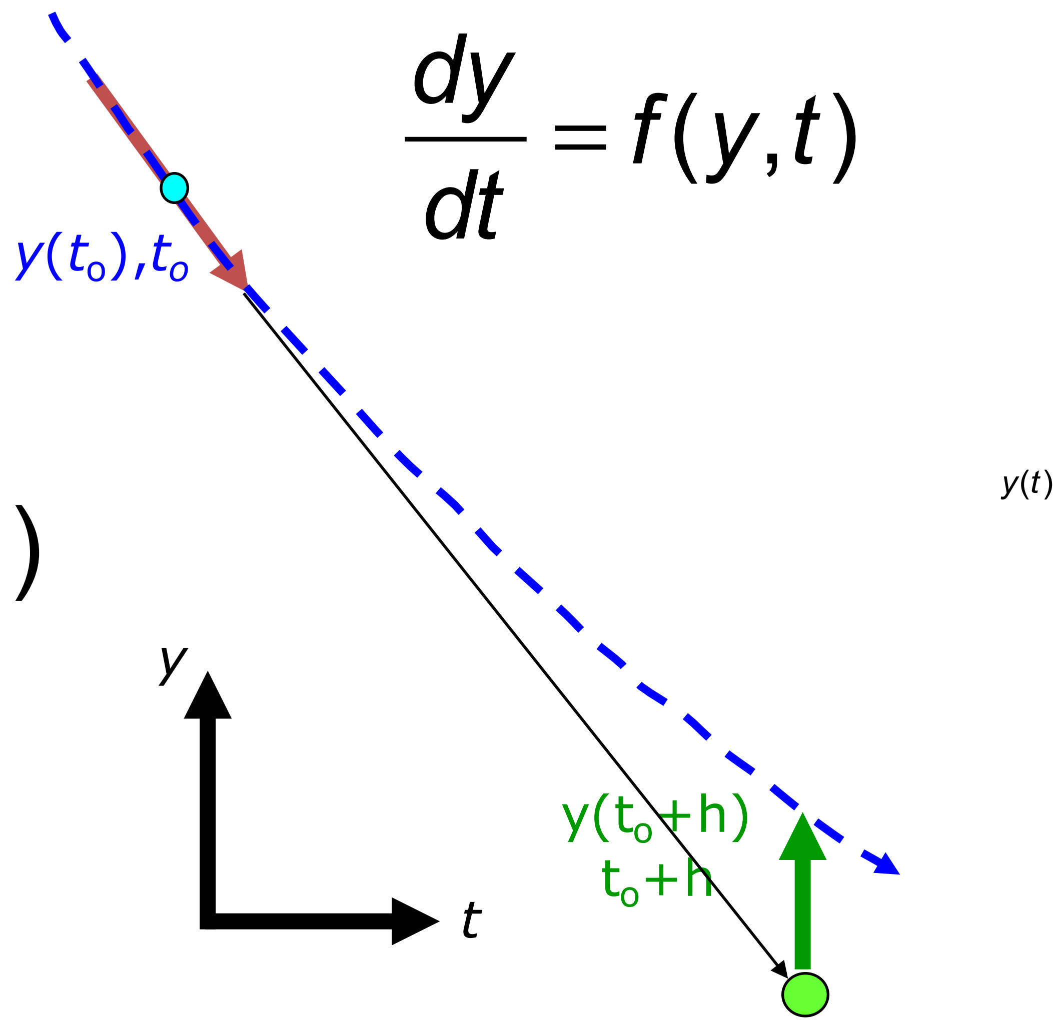
$$k_1 = h\,f(t_n, y_n)$$



1^{rst}-Order Runge Kutta

- And for completeness
  - 1^{rst}-Order R-K is just Euler's Method

$$\frac{dy}{dt} = f(y,t)$$



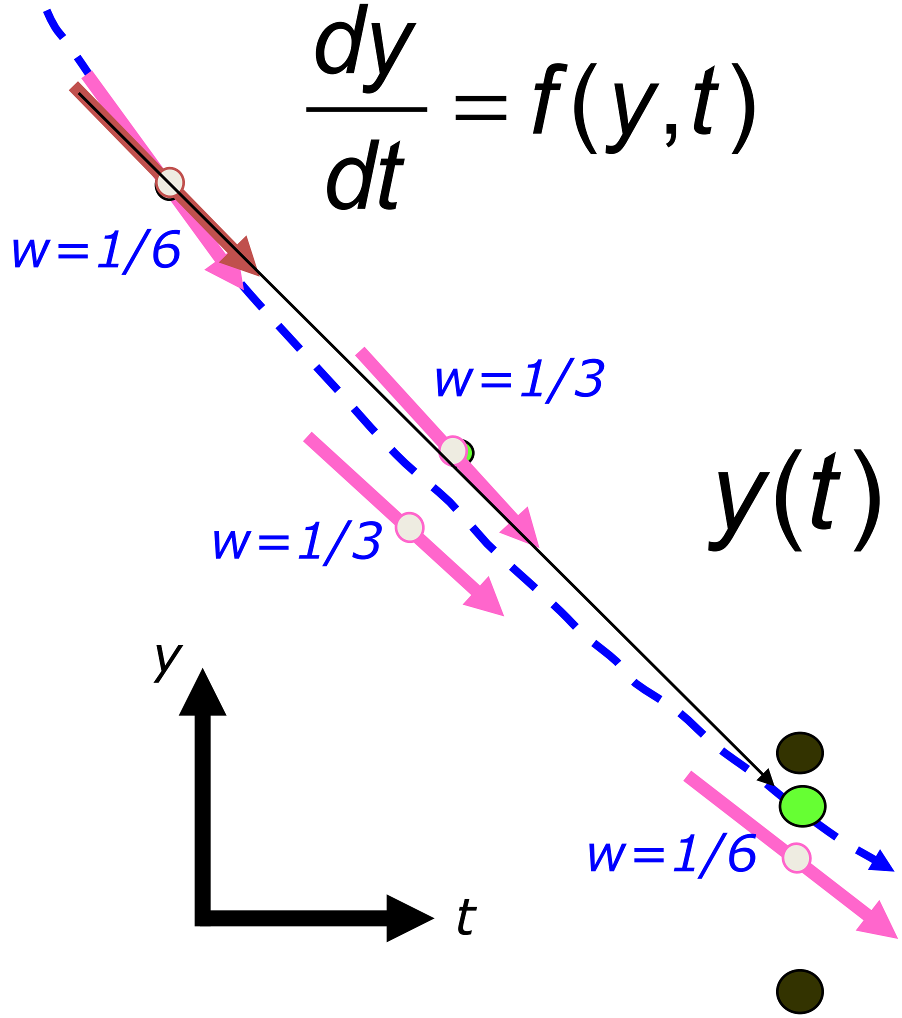
$$y_{n+1} = y_n + k_1 + O(h^2)$$

$$k_1 = h\,f(t_n, y_n)$$

Runge Kutta vs Euler

- Though in this case, we did quite well in going from a Euler approach, through the R-K approaches, higher order approaches are not a guarantee of an improved solution.

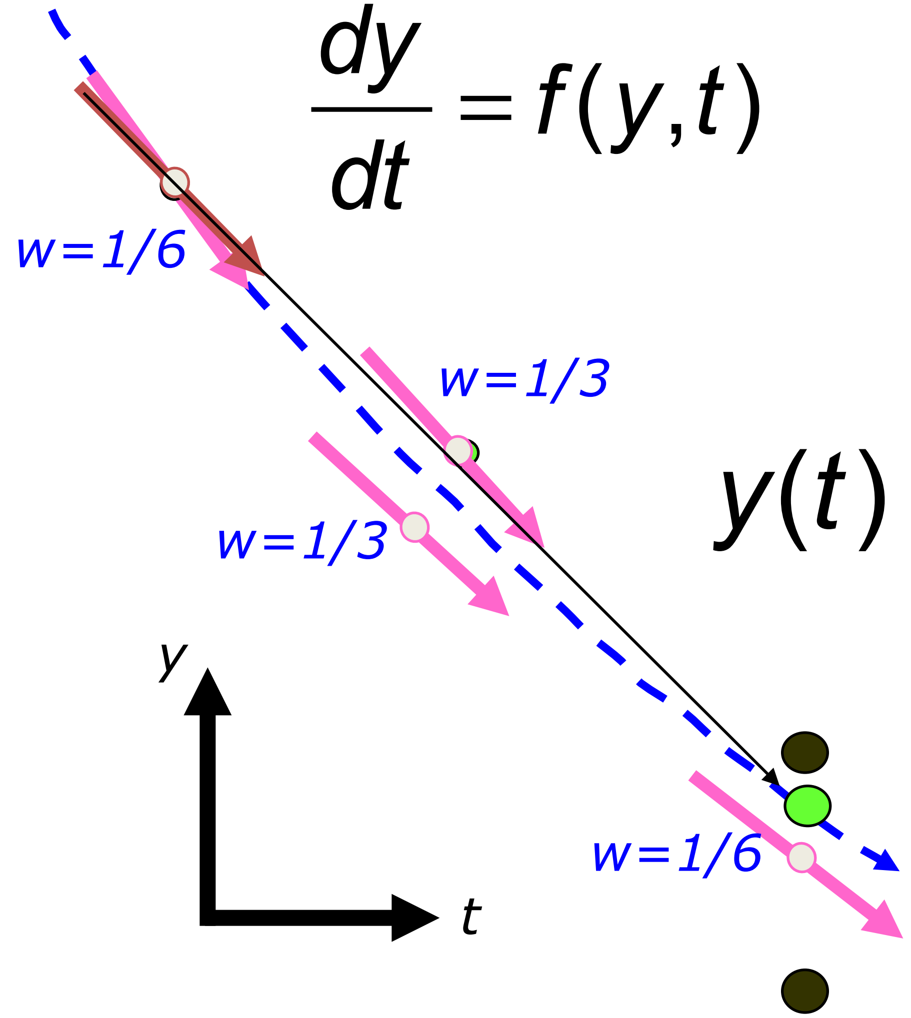
$$\frac{dy}{dt} = f(y,t)$$



Runge Kutta vs Euler

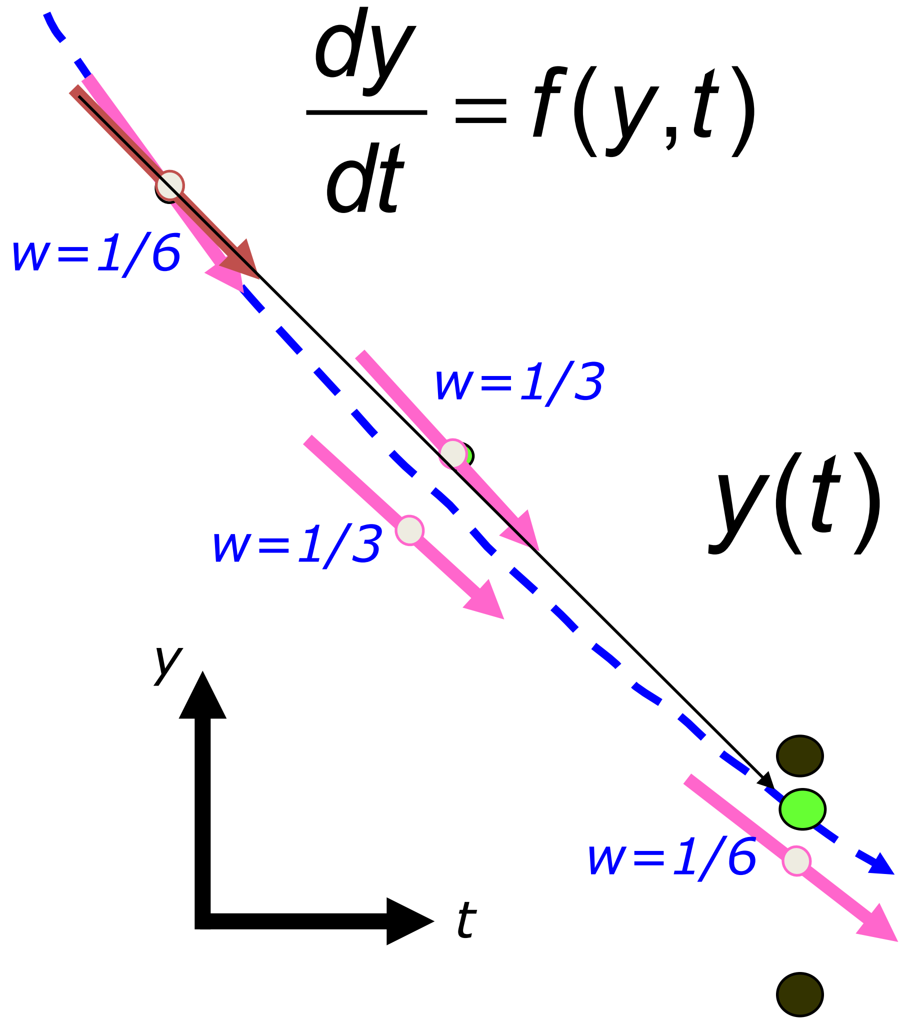
- If you have a smooth continuous function over your time step, R-K should service you fairly well on most occasions if you use the same time step as a Euler approach.
- However, R-K doesn't not liberate you from needing sensible time steps.

$$\frac{dy}{dt} = f(y,t)$$



Runge Kutta vs Euler

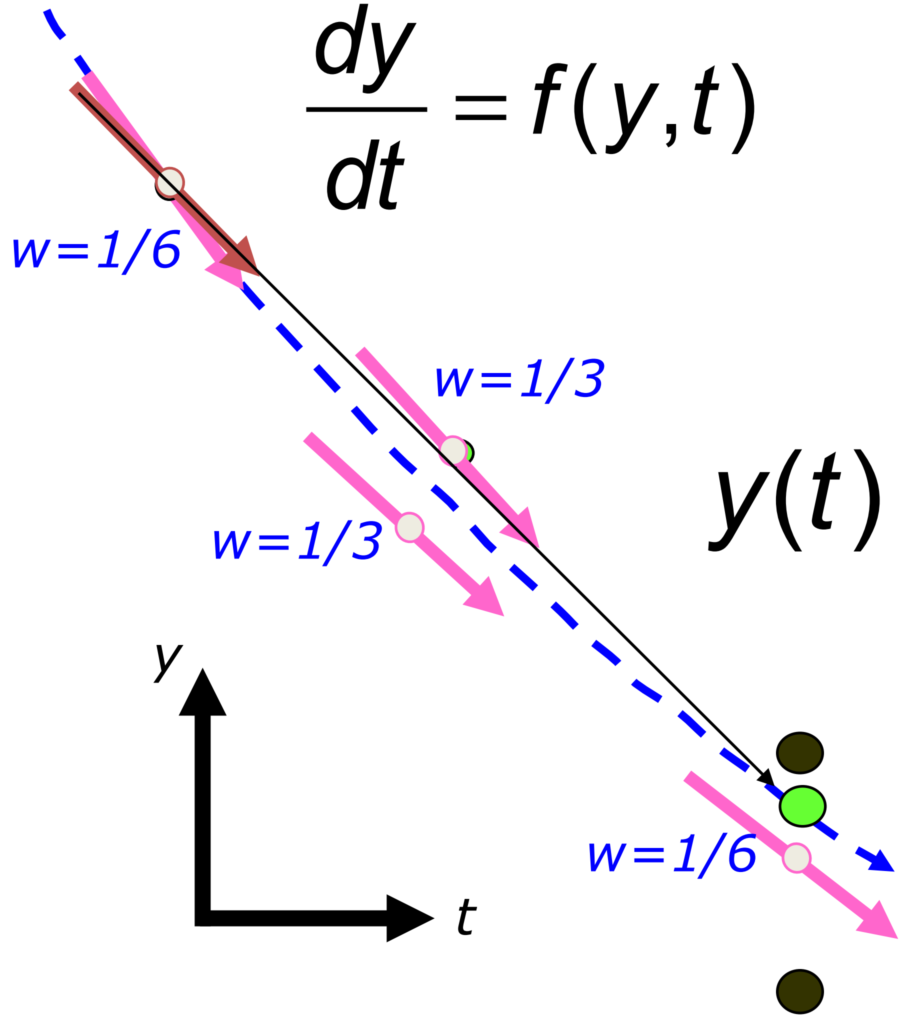
- Stella™ also doesn' t recommend R-K approaches when you use some of their options or when you periodically "shock" your system.
- R-K does provide a good representation of smooth continuous and oscillating systems.



Runge Kutta vs Euler

- But not matter what scheme you use,
- ALWAYS DO A TIME-STEP SENSITIVITY TEST!

$$\frac{dy}{dt} = f(y,t)$$



$y(t)$

Let's Play